

Continuation model predictive control on smooth manifolds

Andrew Knyazev^{*} Alexander Malyshev^{**}

^{*} *Mitsubishi Electric Research Labs (MERL) 201 Broadway, 8th floor,
Cambridge, MA 02139, USA. (e-mail: knyazev@merl.com),
(http://www.merl.com/people/knyazev).*

^{**} *Mitsubishi Electric Research Labs (MERL) 201 Broadway, 8th floor,
Cambridge, MA 02139, USA. (e-mail: malyshev@merl.com)*

Abstract: Model predictive control (MPC) anticipates future events to take appropriate control actions. Nonlinear MPC (NMPC) describes systems with nonlinear models and/or constraints. Continuation MPC, suggested by T. Ohtsuka in 2004, uses Krylov-Newton iterations. Continuation MPC is suitable for nonlinear problems and has been recently adopted for minimum time problems. We extend the continuation MPC approach to a case where the state is implicitly constrained to a smooth manifold. We propose an algorithm for on-line controller implementation and demonstrate its numerical effectiveness for a test problem on a hemisphere.

Keywords: nonlinear model predictive control, Newton-Krylov method, geometric integration, structure-preserving algorithm.

1. INTRODUCTION

Model predictive control (MPC) is exposed, e.g., in Camacho et al. (2004); Grüne et al. (2011). MPC is efficient in industrial applications; see Qin et al. (2003). Numerical aspects of MPC are discussed in Diehl et al. (2009). Our contribution to numerical methods for MPC is further development of the approach by Ohtsuka (2004) and efficient use of the Newton-Krylov approximation for numerical solving the nonlinear MPC problems. Knyazev et al. (2015a) solves a minimum-time problem and investigates preconditioning for the Newton-Krylov method.

The MPC method owes its success to efficient treatment of constraints on control and state variables. The goal of the present note is to draw attention to the important fact that, in addition to the efficient treatment of explicit constraints, MPC can easily incorporate the structure-preserving geometric integration of ordinary differential equations modeling the system.

The idea of structure-preserving numerical methods for differential equations has been employed in numerical analysis since 1950s, when the first computers began to be used in engineering computations. However, its systematic study under the name of *geometric integration of differential equations* has been accomplished within the last 40 years. An accessible introduction to this active research domain is found in Hairer (2011). A more advanced source is Hairer et al. (2006).

Variational formulations of many models described by differential equations lead to the Hamiltonian system dynamics; see Leimkuhler et al. (2004). Such dynamic systems

conserve the energy and possess a special geometric structure called the symplectic structure. Much effort has been spent by numerical analysts to develop the called symplectic numerical methods, which are especially efficient for long-time integration, because they produce qualitatively correct computed solutions. The symplectic methods are superior even in the short-time simulations. Leimkuhler et al. (2004) presents the state-of-the-art symplectic numerical algorithms and supporting theory.

Description of the system dynamics in the form of ordinary differential equations appears twice in MPC: first, within the finite-horizon prediction problem and, secondly, when advancing the current state with the input control computed by the finite-horizon prediction. The latter can be omitted if the states of the system are measured directly by the controller sensors. Furthermore, the application of the geometric structure-preserving integration algorithms from Hairer et al. (2006) or other sources for advancing the current state is straightforward.

The use of the geometric integration during the finite-horizon prediction is less obvious. The main difference between the method from Knyazev et al. (2015a) and the variant of the present paper is in that the latter applies a structure-preserving geometric integration inside the forward recursion.

The rest of the note is organized as follows. Section 2 presents a framework of the finite-horizon prediction problem and expresses its solution in the form of a nonlinear algebraic equation. Geometric integration is incorporated during the elimination of state variables from the KKT conditions. Section 3 discusses how classical integration methods can be used for integration on manifolds. The content of Section 3 is mainly adopted from Hairer (2001) and discusses the local coordinates approach and projec-

^{*} Accepted to the 16th IFAC Workshop on Control Applications of Optimization (CAO'2015), Garmisch-Partenkirchen, Germany, October 6–9, 2015.

tion methods. Section 4 describes a simple test example and necessary formulas for computer implementation. Section 5 shows numerical results and plots.

2. FINITE-HORIZON PREDICTION

Our model finite-horizon control problem, see Knyazev et al. (2015a), along a fictitious time $\tau \in [t, t + T]$ consists in choosing the control $u(\tau)$ and parameter vector p , which minimize the performance index J as follows:

$$\min_{u, p} J,$$

where

$$J = \phi(x(t + T), p) + \int_t^{t+T} L(\tau, x(\tau), u(\tau), p) d\tau$$

subject to the equation for the state dynamics

$$\frac{dx}{d\tau} = f(\tau, x(\tau), u(\tau), p), \quad (1)$$

and the equality constraints for the state x and the control u

$$C(\tau, x(\tau), u(\tau), p) = 0, \quad (2)$$

$$\psi(x(t + T), p) = 0. \quad (3)$$

The initial value condition $x(\tau)|_{\tau=t}$ for (1) is the state vector $x(t)$ of the dynamic system. The control vector $u = u(\tau)|_{\tau=t}$ is used afterwards as an input to control the system at time t . The components of the vector $p(t)$ are parameters of the dynamic system. The horizon time length T may depend on t .

The continuous formulation of the finite-horizon prediction problem stated above is discretized on a time grid τ_i , $i = 0, 1, \dots, N$, through the horizon $[t, t + T]$ partitioned into N time steps of size $\Delta\tau_i = \tau_{i+1} - \tau_i$, and the time-continuous vector functions $x(\tau)$ and $u(\tau)$ are replaced by their indexed values x_i and u_i at the grid points τ_i . The integral of the performance cost J over the horizon is approximated by the rectangular quadrature rule. Equation (1) is approximated by a one-step integration formula, for example, a Runge-Kutta method; cf. Hairer (2001). The discretized optimal control problem is as follows:

$$\min_{u_i, p} \left[\phi(x_N, p) + \sum_{i=0}^{N-1} L(\tau_i, x_i, u_i, p) \Delta\tau_i \right],$$

subject to

$$x_{i+1} = x_i + \Phi_i(\tau_i, x_i, u_i, p) \Delta\tau_i, \quad i = 0, 1, \dots, N-1, \quad (4)$$

$$C(\tau_i, x_i, u_i, p) = 0, \quad i = 0, 1, \dots, N-1, \quad (5)$$

$$\psi(x_N, p) = 0. \quad (6)$$

The function Φ_i is implicitly determined by the structure-preserving method used for numerical integration of (1). We note that $\Phi_i(\tau_i, x_i, u_i, p) = f(\tau_i, x_i, u_i, p) + O(\Delta\tau)$, where $\Delta\tau = \max_i \Delta\tau_i$.

The necessary optimality conditions for the discretized finite horizon problem are obtained by means of the discrete Lagrangian function

$$\begin{aligned} \mathcal{L}(X, U) = & \phi(x_N, p) + \sum_{i=0}^{N-1} L(\tau_i, x_i, u_i, p) \Delta\tau_i \\ & + \lambda_0^T [x(t) - x_0] \\ & + \sum_{i=0}^{N-1} \lambda_{i+1}^T [x_i - x_{i+1} + \Phi_i(\tau_i, x_i, u_i, p) \Delta\tau_i] \\ & + \sum_{i=0}^{N-1} \mu_i^T C(\tau_i, x_i, u_i, p) \Delta\tau_i + \nu^T \psi(x_N, p), \end{aligned}$$

where $X = [x_i \ \lambda_i]^T$, $i = 0, 1, \dots, N$, and $U = [u_i \ \mu_i \ \nu \ p]^T$, $i = 0, 1, \dots, N-1$. Here, λ is the costate vector and μ is the Lagrange multiplier vector associated with constraint (5). The terminal constraint (6) is relaxed by the aid of the Lagrange multiplier ν .

The Hamiltonian function is denoted by

$$\begin{aligned} H(t, x, \lambda, u, \mu, p) = & L(t, x, u, p) \\ & + \lambda^T f(t, x, u, p) + \mu^T C(t, x, u, p). \end{aligned}$$

The necessary optimality conditions are the KKT stationarity conditions: $\mathcal{L}_{\lambda_i} = 0$, $\mathcal{L}_{x_i} = 0$, $i = 0, 1, \dots, N$, $\mathcal{L}_{u_j} = 0$, $\mathcal{L}_{\mu_j} = 0$, $i = 0, 1, \dots, N-1$, $\mathcal{L}_{\nu_k} = 0$, $\mathcal{L}_{p_l} = 0$.

Since $\partial\Phi_i(\tau_i, x_i, u_i, p)/\partial x_i$ is not always available, we suggest to use $\partial f(\tau_i, x_i, u_i, p)/\partial x_i$ instead. This modification is applied in the backward recursion used below.

The KKT conditions are reformulated in terms of a mapping $F[U, x, t]$, where the vector U combines the control input u , the Lagrange multiplier μ , the Lagrange multiplier ν , and the parameter p , all in one vector:

$$U(t) = [u_0^T, \dots, u_{N-1}^T, \mu_0^T, \dots, \mu_{N-1}^T, \nu^T, p^T]^T.$$

The vector argument x in $F[U, x, t]$ denotes the current measured or estimated state vector, which serves as the initial vector x_0 in the following procedure.

- (1) Starting from the current measured or estimated state x_0 , compute x_i , $i = 0, 1, \dots, N-1$, by the forward recursion

$$x_{i+1} = x_i + \Phi_i(\tau_i, x_i, u_i, p) \Delta\tau_i.$$

Then starting from

$$\lambda_N = \frac{\partial\phi^T}{\partial x}(x_N, p) + \frac{\partial\psi^T}{\partial x}(x_N, p)\nu$$

compute the costates λ_i , $i = N-1, \dots, 0$, by the backward recursion

$$\lambda_i = \lambda_{i+1} + \frac{\partial H^T}{\partial x}(\tau_i, x_i, \lambda_{i+1}, u_i, \mu_i, p) \Delta\tau_i.$$

- (2) Calculate $F[U, x, t]$, using just obtained x_i and λ_i , as

$$\begin{aligned}
& F[U, x, t] \\
& \left[\begin{array}{c} \frac{\partial H^T}{\partial u}(\tau_0, x_0, \lambda_1, u_0, \mu_0, p) \Delta \tau_0 \\ \vdots \\ \frac{\partial H^T}{\partial u}(\tau_i, x_i, \lambda_{i+1}, u_i, \mu_i, p) \Delta \tau_i \\ \vdots \\ \frac{\partial H^T}{\partial u}(\tau_{N-1}, x_{N-1}, \lambda_N, u_{N-1}, \mu_{N-1}, p) \Delta \tau_{N-1} \\ C(\tau_0, x_0, u_0, p) \Delta \tau_0 \\ \vdots \\ C(\tau_i, x_i, u_i, p) \Delta \tau_i \\ \vdots \\ C(\tau_{N-1}, x_{N-1}, u_{N-1}, p) \Delta \tau_{N-1} \\ \psi(x_N, p) \\ \frac{\partial \phi^T}{\partial p}(x_N, p) + \frac{\partial \psi^T}{\partial p}(x_N, p) \nu \\ + \sum_{i=0}^{N-1} \frac{\partial H^T}{\partial p}(\tau_i, x_i, \lambda_{i+1}, u_i, \mu_i, p) \Delta \tau_i \end{array} \right] \cdot
\end{aligned}$$

The equation with respect to the unknown vector $U(t)$

$$F[U(t), x(t), t] = 0 \quad (7)$$

gives the required necessary optimality conditions that are solved on the controller in real time. All details about solving (7) by the Newton-Krylov method are found in Knyazev et al. (2015a). We only recall that our method uses GMRES iterations for solving linear systems with the Jacobian matrix F_U . To accelerate the convergence of GMRES iterations, the matrix F_U is computed exactly at some time instances and then used as a preconditioner between these time instances.

3. GEOMETRIC INTEGRATION

This section gives a short explanation of geometric integration used in our illustrative example. We consider the ordinary differential equations (1), which describe the system dynamics. Suppose that a property $L(x(t)) = \text{const}$ is fulfilled on each solution $x(t)$ of (1), where the constant const depends on the solution. If the numerical method $x_{i+1} = x_i + \Phi_i(\tau_i, x_i, u_i, p) \Delta \tau_i$ satisfies the same property $L(x_i(\tau)) = \text{const}$, we refer to it as a structure preserving method. A notorious example is given by the spheres $L(x(t)) = \|x\|_2 = \text{const}$.

The article by Hairer (2001) “illustrates how classical integration methods for differential equations on manifolds can be modified in order to preserve certain geometric properties of the exact flow. Projection methods as well as integrators based on local coordinates are considered.”

We adopt the simplest method from Hairer (2001), which is called the *method of local coordinates* there. When it is feasible, it is the most accurate of all structure preserving methods. Let a manifold \mathcal{M} contain the solution of $\dot{x} = f(x)$, which we want to compute. Assume that $\alpha: U \rightarrow R^n$ is a local parametrization of \mathcal{M} near the state $x_i = \alpha(z_i)$.

The coordinate change $x = \alpha(z)$ transforms the differential equation $\dot{x} = f(x)$ into

$$\alpha'(z)\dot{z} = f(\alpha(z)). \quad (8)$$

This is an over-determined system of differential equations because the dimension of z is less than $n = \dim x$. However, $f(x)$ is tangent to \mathcal{M} by assumption, therefore, (8) is equivalent to a system

$$\dot{z} = \beta(z), \quad z(\tau_i) = z_i. \quad (9)$$

If the transfer from (8) to (9) is easy for implementation, then the method of local coordinates is feasible.

The principal idea of the method of local coordinates is to perform one step of a numerical method applied to (9) and to map the result via the transformation α back to the manifold. One step $x_i \mapsto x_{i+1}$ of the resulting algorithm is implemented as follows.

Algorithm 1 (Local coordinates approach)

- choose a local parametrization α and compute z_i from $x_i = \alpha(z_i)$;
- compute $\hat{z}_{i+1} = z_i + \Phi_i(z_i) \Delta \tau_i$, the result of the one-step method applied to (9);
- define the numerical solution by $x_{i+1} = \alpha(\hat{z}_{i+1})$.

3.1 Symmetric integration methods

Mechanical systems often obey the Hamiltonian structure, and numerical methods for such system must be symmetric or time-reversible in order to preserve geometric properties of the Hamiltonian systems. Assume that a one-step numerical method applied to a system $\dot{x} = f(t, x)$ over the interval $[t_i, t_{i+1}]$ produces the state x_{i+1} from the state x_i . The time-reversibility of the numerical method means that its application to the time-reversed system $-\dot{x} = f(-t, x)$ over the interval $[-t_{i+1}, -t_i]$ produces x_i from x_{i+1} . This is a case for the trapezoidal rule, but not for the explicit Euler method.

Hairer (2001) shows that symmetric methods perform qualitatively better for integration over long time intervals. But most of the commonly used techniques for solving differential equations on manifolds destroy the symmetry of underlying method. To restore the symmetry, additional modifications of numerical methods are necessary.

We illustrate restoration of the symmetry for another standard technique of geometric integration on manifolds called the *projection methods*. For an ordinary differential equation $\dot{y} = f(y)$, the one step integration $y_n \mapsto y_{n+1}$ proceeds as follows.

Algorithm 2 (Standard projection method)

- compute $\hat{y}_{n+1} = \Phi_h(y_n)$ by any numerical integrator Φ_h applied to $\dot{y} = f(y)$, e.g. by a Runge-Kutta method;
- project \hat{y}_{n+1} orthogonally onto the manifold \mathcal{M} to obtain $y_{n+1} \in \mathcal{M}$.

The standard projection method is illustrated in Figure 1. The projection destroys the symmetry of Φ_h if it is available. A symmetry-restoration algorithm is developed in Hairer (2000). The idea is to perturb the vector y_n

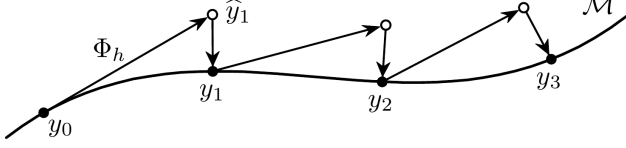


Fig. 1. The standard projection (Hairer, 2001, Fig. 3.1)

before applying a symmetric one-step method such that the final projection is of the same size as the perturbation.

Algorithm 3 (Symmetric projection method)

- $\tilde{y}_n = y_n + G^T(y_n)\mu$, where $g(y_n) = 0$;
- $\hat{y}_{n+1} = \Phi_h(\tilde{y}_n)$ (symmetric one-step method for equation $\dot{y} = f(y)$);
- $y_{n+1} = \hat{y}_{n+1} + G^T(y_{n+1})\mu$ with vector μ such that $g(y_{n+1}) = 0$.

Here, $G(y) = g'(y)$ denotes the Jacobian of $g(y)$, if the manifold is given by the condition $\mathcal{M} = \{y | g(y) = 0\}$. It is important to choose the same vector μ in the perturbation as in the projection.

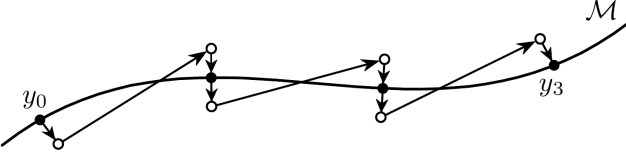


Fig. 2. The symmetric projection (Hairer, 2001, Fig. 3.3)

4. TEST PROBLEM

We consider the minimum-time motion over the unit upper hemisphere from a state (x_0, y_0, z_0) to a state (x_f, y_f, z_f) with inequality constraints.

The system dynamics is governed by the system of differential equations

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} z \cos u \\ z \sin u \\ -x \cos u - y \sin u \end{bmatrix}. \quad (10)$$

The control variable u is subject to an inequality constraint. Namely, the control u always stays within the band $c_u - r_u \leq u \leq c_u + r_u$. Following Ohtsuka (2004) we introduce a slack variable u_s and replace the inequality constraint by the equality constraint

$$C(u, u_d) = (u - c_u)^2 + u_s^2 - r_u^2 = 0.$$

In order to guarantee that the state passes through the point (x_f, y_f, z_f) at time $t = t_f$, we impose three terminal constraints of the form

$$\psi(x, y, z, p) = \begin{bmatrix} x - x_f \\ y - y_f \\ z - z_f \end{bmatrix} = 0.$$

The objective is to minimize the cost function

$$J = \phi(p) + \int_{t_0}^{t_f} L(x, y, z, u, u_s, p) dt',$$

where

$$\phi(p) = p = t_f - t_0, \quad L(x, y, z, u, u_s, p) = -w_s u_s.$$

The term $\phi(p)$ is responsible for the shortest time to destination, and the function L serves for stabilization of the slack variable u_s .

System (10) possesses the first integral

$$x^2 + y^2 + z^2 = \text{const.}$$

Our initial condition lies on the unit sphere $x_0^2 + y_0^2 + z_0^2 = 1$. Therefore, the manifold \mathcal{M} is the unit sphere with the center at the origin. To simplify the implementation, we choose all parameters such that the trajectories lie on the upper hemisphere $z \geq 0$. The natural local coordinates in this case are x and y , and the local parametrization is given by $\alpha(x, y) = [x, y, \sqrt{1 - x^2 - y^2}]^T$. The system (9) for (10) will be

$$\frac{d}{dt} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sqrt{1 - x^2 - y^2} \cos u \\ \sqrt{1 - x^2 - y^2} \sin u \end{bmatrix}. \quad (11)$$

For this particular example and choice of local coordinates the structure preserving geometric integration solver consists of the explicit Euler method for the components x and y and the formula $z = \sqrt{1 - x^2 - y^2}$ for the component z .

For convenience, we change the time variable t within the horizon by the new time $\tau = (t - t_0)/(t_f - t_0)$, which runs over the interval $[0, 1]$.

The corresponding discretized finite-horizon problem on a uniform grid τ_i in the local coordinates x, y comprises the following data structures and computations:

- $\tau_i = i\Delta\tau$, where $i = 0, 1, \dots, N$, and $\Delta\tau = 1/N$;
- the participating variables are the state $\begin{bmatrix} x_i \\ y_i \end{bmatrix}$, the costate $\begin{bmatrix} \lambda_{1,i} \\ \lambda_{2,i} \end{bmatrix}$, the control $\begin{bmatrix} u_i \\ u_{si} \end{bmatrix}$, the Lagrange multipliers μ_i and $\begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix}$;
- the state is governed by the model equation
$$\begin{cases} x_{i+1} = x_i + \Delta\tau \left(p \sqrt{1 - x_i^2 - y_i^2} \cos u_i \right), \\ y_{i+1} = y_i + \Delta\tau \left(p \sqrt{1 - x_i^2 - y_i^2} \sin u_i \right), \end{cases}$$
where $i = 0, 1, \dots, N - 1$;
- the costate is computed by the backward recursion
$$(\lambda_{1,N} = \nu_1, \lambda_{2,N} = \nu_2)$$

$$\begin{cases} \lambda_{1,i} = \lambda_{1,i+1} - \Delta\tau p \frac{x_i}{\sqrt{1 - x_i^2 - y_i^2}} \\ \quad \cdot (\cos u_i \lambda_{1,i+1} + \sin u_i \lambda_{2,i+1}), \\ \lambda_{2,i} = \lambda_{2,i+1} - \Delta\tau p \frac{y_i}{\sqrt{1 - x_i^2 - y_i^2}} \\ \quad \cdot (\cos u_i \lambda_{1,i+1} + \sin u_i \lambda_{2,i+1}), \end{cases}$$
where $i = N - 1, N - 2, \dots, 0$;
- the nonlinear equation $F(U, x_0, t_0) = 0$, where

$$U = [u_0, \dots, u_{N-1}, u_{s,0}, \dots, u_{s,N-1}, \mu_0, \dots, \mu_{N-1}, \nu_1, \nu_2, p],$$

has the following rows from the top to bottom:

$$\begin{cases} \Delta\tau p [\sqrt{1 - x_i^2 - y_i^2} (-\sin u_i \lambda_{1,i+1} + \cos u_i \lambda_{2,i+1}) \\ \quad + 2(u_i - c_u) \mu_i] = 0 \end{cases}$$

$$\begin{aligned}
& \{ \Delta \tau p [2\mu_i u_{si} - w_s] = 0 \\
& \{ \Delta \tau p [(u_i - c_u)^2 + u_{si}^2 - r_u^2] = 0 \\
& \begin{cases} x_N - x_f = 0 \\ y_N - y_f = 0 \end{cases} \\
& \left\{ \Delta \tau \left\{ \sum_{i=0}^{N-1} \sqrt{1 - x_i^2 - y_i^2} (\cos u_i \lambda_{1,i+1} + \sin u_i \lambda_{2,i+1}) \right. \right. \\
& \quad \left. \left. + \mu_i [(u_i - c_u)^2 + u_{si}^2 - r_u^2] - w_s u_{si} \right\} + 1 = 0. \right.
\end{aligned}$$

5. NUMERICAL RESULTS

In our numerical experiments, we use the method of the local coordinates for geometric integration on the unit upper hemisphere.

The number of grid points on the horizon is $N = 20$, the time step of the dynamic system is $\Delta t = 0.00625$, the end points of the computed trajectory are $(x_0, y_0) = (-0.5, 0.5)$ and $(x_f, y_f) = (0.5, 0)$. The constants of the inequality constraint for the control are $c_u = 0.5$ and $r_u = 0.1$. Other parameters are $h = 10^{-8}$ and $w_s = 0.005$.

We use the GMRES method without restarts. The number of GMRES iterations does not exceed $k_{\max} = 20$, and the absolute tolerance of the GMRES iterations is 10^{-5} .

Preconditioning of GMRES accelerates its convergence as demonstrated on Figures 6 and 7. We used a simple preconditioning strategy as follows. The exact Jacobian F_U is computed periodically at time instances with the period 0.2 seconds. Then the LU factorization of the Jacobian is used as the preconditioner until the next time when it is recalculated.

The value of U at t_0 is approximated by the Matlab function `fsolve` with a special initial guess.

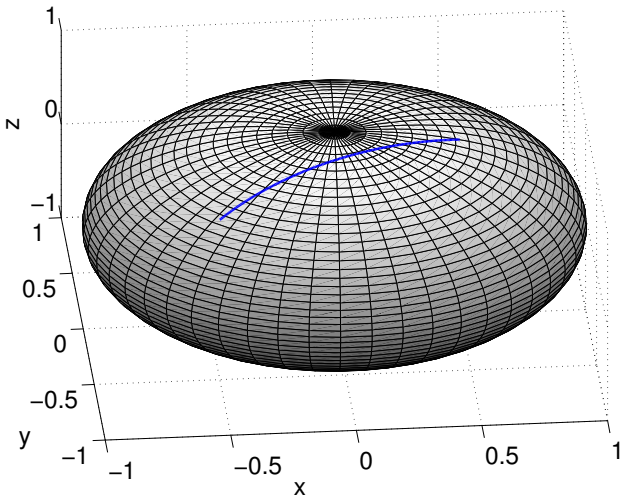


Fig. 3. 3D plot of the trajectory on the unit hemisphere. Time to destination equals 1.2332.

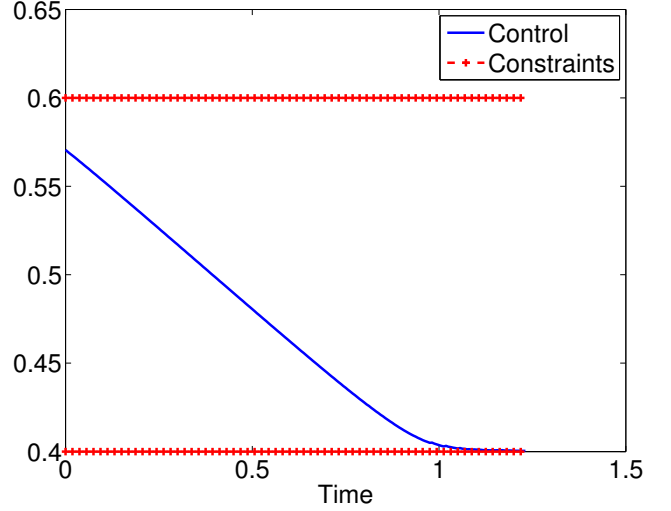


Fig. 4. The constrained control

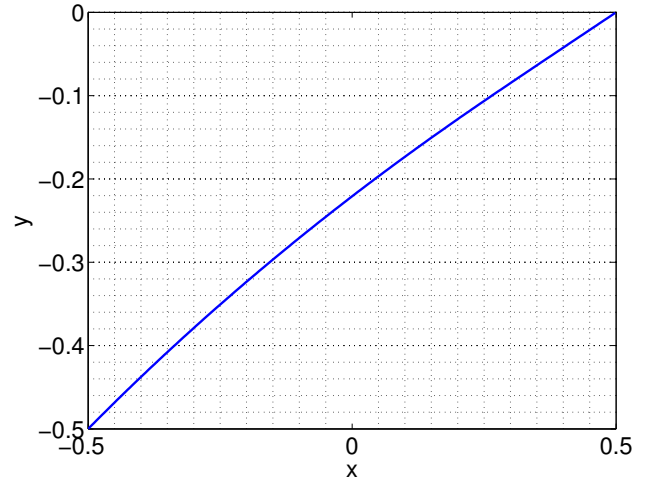


Fig. 5. 2D projection of the trajectory

6. CONCLUSION

Model predictive control is efficient in dealing with the constraints on the control and state variables. When the state space of a system lies on a manifold, it may be profitable to use numerical methods which inherit this property. For example, numerical algorithms preserving the symplectic structure of Hamiltonian dynamical systems are much superior in accuracy compared to the conventional algorithms. In this note, we show how to incorporate simple modifications of classical integration methods into our numerical approach to MPC obtaining an efficient structure-preserving NMPC method. Other structure-preserving algorithms can be used similarly.

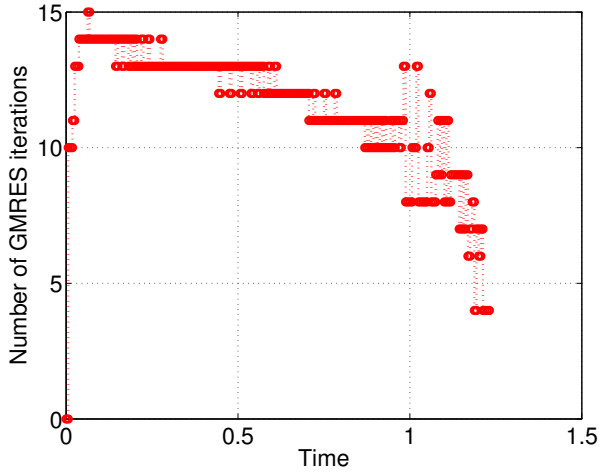


Fig. 6. Number of GMRES iterations without preconditioning

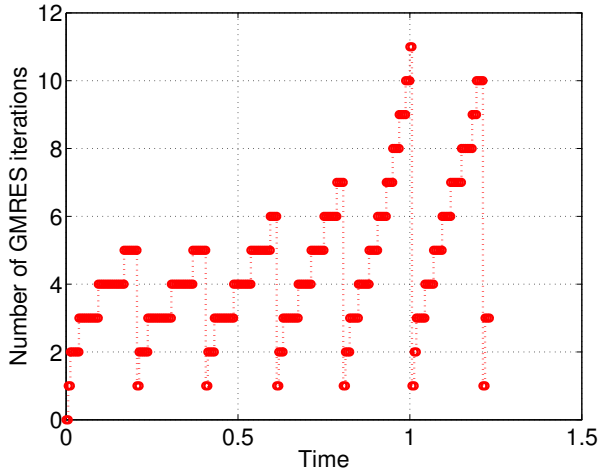


Fig. 7. Number of GMRES iterations with preconditioning

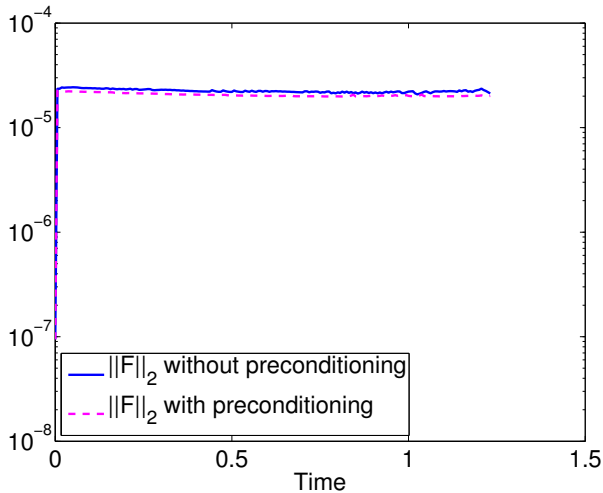


Fig. 8. Norm of the residual $\|F\|_2$

REFERENCES

- E. F. Camacho and C. Bordons. *Model Predictive Control, 2nd ed.* Springer, Heidelberg, 2004.
- M. Diehl, H. J. Ferreau, and N. Haverbeke. Efficient numerical methods for nonlinear MPC and moving horizon estimation. L. Magni et al. (Eds.): *Nonlinear Model Predictive Control*, LNCIS 384, pp. 391–417, Springer, Heidelberg, 2009.
- L. Grüne and J. Pannek. *Nonlinear Model Predictive Control. Theory and Algorithms.* Springer, London, 2011.
- E. Hairer. Symmetric projection methods for differential equations on manifolds. *BIT*, 40:726–734, 2000.
- E. Hairer. Geometric integration of ordinary differential equations on manifolds. *BIT*, 41:996–1007, 2001.
- E. Hairer, G. Wanner, and Ch. Lubich. *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations.* Springer, Berlin Heidelberg, 2006.
- E. Hairer. *Solving Differential Equations on Manifolds.* Université de Genève, Genève, 2011. <http://www.unige.ch/~hairer/poly-sde-mani.pdf>
- A. Knyazev, Y. Fujii, and A. Malyshev. Preconditioned Continuation Model Predictive Control. *SIAM Conf. Control. Appl., July 8–10, 2015, Paris, France*, pp. 1–8, 2015.
- B. Leimkuhler and S. Reich. *Simulating Hamiltonian Dynamics.* Cambridge University Press, 2004.
- T. Ohtsuka. A Continuation/GMRES method for fast computation of nonlinear receding horizon control. *Automatica*, 40:563–574, 2004.
- S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Eng. Practice*, 11: 733–764, 2003.